The Purdue Model
How the Industry Stripped a Methodology Down to a Cartoon, and Why It's Time to Bring It Back
≋RIVER

The Purdue Model Isn't Dead

River Caudle

CSO, River Risk Partners | Industrial Cybersecurity, Risk & Production Loss Prevention | Nuclear, Energy & Critical Infrastructure | Author &

January 17, 2026

How the Industry Stripped a Methodology Down to a Cartoon, and Why It's Time to Bring It Back

For the last decade, OT security professionals have been locked in a circular debate: Is the Purdue Model still relevant? Is it dead? Does it need to be replaced?

Here's the problem with that debate: almost no one is talking about the actual Purdue Model.

What gets discussed on LinkedIn, debated at conferences, and referenced in vendor whitepapers is a simplified diagram - a stack of boxes labeled Level 0 through Level 5. That diagram has become so ubiquitous that the industry forgot it was ever anything else.

But the Purdue Enterprise Reference Architecture (PERA) was never just a network diagram. It was a comprehensive methodology for enterprise integration, one that remains remarkably relevant to the challenges we face today, if we bother to actually read it.

This article is an attempt to correct the record.

Part 1: The Great Reduction

Sometime in the early 2000s, a transformation happened. The Purdue Enterprise Reference Architecture - a methodology developed over years at Purdue University's Consortium for Computer Integrated Manufacturing - got compressed into a single image.

That image showed hierarchical levels: physical processes at the bottom, business systems at the top, with control, supervisory, and operations layers in between. It was useful shorthand. It helped people visualize where different systems lived in the enterprise.

And then the shorthand became the whole damn thing.

Vendors adopted the diagram to sell segmentation products. Security frameworks referenced "the Purdue Model" as if it were self-explanatory. Consultants built network architectures by drawing firewalls between the levels and calling it done.

What got lost:

- **The lifecycle framework.** PERA wasn't a snapshot, it was a methodology for understanding how enterprises evolve from initial concept through operations to eventual dissolution. The architecture at each phase looks different.
- **The three parallel architectures.** PERA defined three concurrent views: the Facilities Architecture (physical plant), the Human and Organizational Architecture (people, roles, procedures), and the Control and Information Systems Architecture (automation). You can't understand industrial operations by looking at only one.
- **Functional decomposition.** PERA provided methods for breaking down enterprise functions before selecting technology. The question was "what does this function need to accomplish?" before "what product should we buy?"
- **The lines of automation.** PERA included concepts for determining where automation was appropriate and where human involvement was necessary-not as a technology limitation, but as a design principle.

The industry took a methodology built on careful functional analysis and turned it into a network topology template. Then, when that template proved insufficient for modern challenges, the critics declared the whole thing dead.

They were attacking a strawman.

Part 2: What PERA Actually Contains

The original PERA methodology, as developed by Theodore Williams and the Industry-Purdue University Consortium, was documented in detail. It remains freely available at **pera.net**, maintained by Gary Rathwell and a community of practitioners who never forgot what it actually said.

Here's what's in the box:

The Enterprise Lifecycle Framework

PERA tracks an enterprise through phases: identification, concept, definition, preliminary design, detailed design, construction, operations, and dissolution. At each phase, the three architectures (Facilities, Human/Organizational, and Information Systems) develop in parallel.

This matters because most industrial security assessments treat the plant as a static artifact. PERA reminds us that facilities evolve, and the architecture appropriate for initial operations may not serve a plant that has been expanded, modified, and patched for thirty years.

Control and Information Architecture Diagrams (CIAD/CIND)

PERA standardized diagramming conventions for information systems that parallel the Process Flow Diagrams (PFDs) and Piping & Instrumentation Diagrams (P&IDs) used for physical plant design.

The CIAD is a high-level view developed during conceptual engineering. The CIND adds detail during preliminary engineering, showing backup systems, network paths, and-critically-systems requiring higher cybersecurity protection levels.

These aren't network diagrams in the IT sense. They're engineering documents that show information flows and processing nodes as integral parts of the plant design, developed in parallel with the physical process design.

The Levels as Functional Hierarchy

Yes, PERA defined levels. But the levels were functional descriptions, not network prescriptions:

- **Level 0:** The physical process itself-actual chemistry, physics, and material transformations. You can't patch thermodynamics. There is no attack surface here.
- **Level 1:** Sensing and actuation-the instruments that measure and manipulate Level 0. These are electronic devices with firmware, configuration, and calibration. They have attack surface.
- **Level 2:** Control-regulatory loops, interlocks, and sequence control.
- **Level 3:** Operations and production management-scheduling, optimization, and coordination.
- **Level 4:** Business logistics-supply chain, resource planning, financial systems.

The levels describe what functions exist and how they relate hierarchically. They don't dictate where you put firewalls.

Maximum and Minimum Lines of Automation

PERA introduced the concept that not everything should be automated, and not everything can be. The "maximum line of automation" represents the boundary beyond which current technology cannot reliably perform. The "minimum line" represents the boundary below which automation is economically or practically necessary.

Between these lines lies a design space where human or automated implementation is a choice-one that should be made deliberately, not by default.

This is the opposite of the "connect everything" mentality that drives most digital transformation initiatives.

Part 3: Consequence-Based Thinking

Here's the reframe that matters: PERA is a consequence framework.

When you understand the levels as functional hierarchy, you recognize that each level carries different consequences when something goes wrong:

- **Level 0: Physics** This is the process itself. The reaction, the flow, the heat transfer. Consequences here are governed by chemistry and thermodynamics. You cannot compromise Level 0-but you can lose visibility into it or lose the ability to control it.
- **Level 1: Sensing and Actuation** These are the devices that observe and act on Level 0-transmitters, analyzers, valves, actuators. They are electronic devices with firmware and configuration. They have attack surface. Failures here mean you don't know what's happening (bad measurement) or can't respond (bad actuation). Consequences are immediate and physical. Response time is measured in milliseconds. There is no retry.
- **Level 2: Control** Failures affect process stability. Loops oscillate, sequences fault, interlocks trip. The process degrades or shuts down. Consequences are operational and potentially physical.
- **Level 3: Operations** Failures affect production coordination. Schedules slip, batches are lost, resources are misallocated. Consequences are economic and operational.
- **Level 4: Business** Failures affect planning and logistics. Financial systems miscalculate, supply chains are disrupted, reports are wrong. Consequences are economic and reputational.

The gradient is clear: as you move down the hierarchy, latency requirements tighten and consequence severity increases. Level 4 can tolerate seconds or minutes of latency and survives bad data gracefully. Level 1 operates in milliseconds and bad commands cause immediate physical harm.

This is why the question isn't "which firewall sits between Level 3 and Level 4?"

The question is: **"What happens when this connection fails, degrades, or is actively abused?"**

When you ask that question honestly, you realize that the consequence profile of a Level 1 device is fundamentally incompatible with the reliability assumptions of cloud connectivity. Not because the cloud is insecure-but because the consequences of interruption or manipulation at Level 1 are categorically different from the consequences at Level 4.

The Purdue levels aren't a network design template. They're a framework for understanding why different parts of the enterprise require different approaches to connectivity, availability, and trust.

A Note on Levels 0/1

Some practitioners put sensors at Level 0. This is a mistake. It conflates the measurement with the thing being measured, and obscures the fact that a temperature transmitter is a device that can be compromised, misconfigured, or spoofed. Level 0 is immutable physics. Level 1 is where security starts to matter.

These devices were often designed without authentication or integrity verification - the attack surface isn't just network-accessible, it's architecturally unprotected.

Part 4: Why "Purdue Is Dead" Is Wrong

The case against Purdue usually goes something like this:

"The Purdue Model was designed for a world of isolated plants. Modern operations require cloud connectivity, remote monitoring, and real-time data integration. The model's rigid hierarchical boundaries prevent digital transformation."

Let's examine this:

"Designed for isolated plants" PERA was developed to support Computer Integrated Manufacturing. It was literally a methodology for integration. The goal was never isolation-it was structured, consequence-aware integration.

"Modern operations require cloud connectivity" Some do. The question is which operations, for what functions, with what consequence profiles. PERA provides the framework to answer those questions. "The cloud exists" is not an argument for connecting Level 1 sensors to it.

"Rigid hierarchical boundaries" The hierarchy isn't rigid-it's functional. Functions at Level 1 have different characteristics than functions at Level 4. That's not a limitation of the model; that's physics.

"Prevents digital transformation" PERA doesn't prevent transformation. It asks: transformation of what, with what consequence profile, and with what failure modes?

The critics aren't wrong that the cartoon version of Purdue is insufficient. A diagram with firewalls between levels doesn't address modern integration challenges. But the solution isn't to abandon hierarchical thinking-it's to engage with the actual methodology that produced the hierarchy.

The people declaring Purdue dead have, in most cases, never read PERA.

Part 5: PERA+ and What's Being Updated

The PERA methodology isn't frozen in 1992. A community of practitioners continues to develop and extend it, maintaining the open framework at **pera.net**.

Current development work includes:

Cybersecurity Integration

The original PERA predated modern cybersecurity concerns, but the framework accommodates them naturally. Security Levels (SL1-SL4 from IEC 62443) map onto PERA's consequence hierarchy. The CIND diagrams explicitly support notation for systems requiring higher protection levels.

Work is ongoing to develop guidance for applying PERA concepts to cybersecurity program development, helping owner-operators understand how the hundreds of requirements in IEC 62443 map onto their specific facilities.

Operational Technology Clarification

There's persistent confusion about the relationship between IT, OT, and industrial control systems. PERA provides clear definitions:

- **ACS (Automation and Control Systems):** Industrial devices including sensors, actuators, HMIs, and SCADA systems. Typically Levels 0-3.
- **IT Systems:** Systems that analyze data and present information to humans. Do not directly control plant equipment.
- **OT Systems:** Systems involving infrastructure and expertise of both ACS and IT. May indirectly influence plant operations, but according to PERA principles, should accomplish this only via ACS systems.

That last point is critical: PERA establishes that business systems should not directly command physical processes. The hierarchy isn't arbitrary-it's a consequence firewall.

Modern Diagram Standards

The CIAD/CIND diagramming conventions are being updated to address modern system architectures while maintaining the engineering discipline that made them useful. The goal is diagrams that integrate into existing engineering workflows alongside PFDs and P&IDs.

Educational Resources

A significant barrier to PERA adoption is simply awareness. Most OT professionals have never seen the original documentation. Ongoing work includes training materials, walkthrough guides, and modules that explain PERA concepts without requiring a week-long course.

Part 6: Industrial Independence as PERA's Natural Conclusion

If you take PERA seriously-if you actually work through consequence-based thinking about where automation belongs, where human involvement is necessary, and what failure modes matter-you arrive at a position that might be called Industrial Independence.

Industrial Independence is not isolation. It's the principle that industrial operations should be able to function without continuous external connectivity. Not because connectivity is bad, but because dependency on connectivity creates consequence profiles that are often unacceptable for critical infrastructure.

Consider:

Dependency Inversion When a Level 1 sensor requires cloud connectivity to function, you've inverted the consequence hierarchy. A Level 4 system failure (internet outage, cloud service interruption, authentication system compromise) now cascades to Level 1 consequences (process disruption, safety system blindness).

PERA's hierarchy exists precisely to prevent this inversion.

Operational Resilience Industrial facilities were designed to run independently. The control room has the information and authority to operate the plant. Remote connectivity was an enhancement, not a requirement.

Somewhere along the way, we built operational models that assume constant connectivity. When the connection drops, the operation degrades. This is architectural debt-and PERA's functional decomposition would have caught it at the design phase.

Attack Surface as Consequence Multiplier Every connection is a potential attack path. PERA's consequence framework makes the risk calculation explicit: a compromised connection at Level 4 affects business systems. A compromised connection at Level 1 affects physics.

The question isn't "how do we secure connectivity to Level 1?" The question is "why does Level 1 need external connectivity at all?"

Industrial Independence doesn't mean never connecting anything. It means asking the consequence question first, and being willing to accept "you shouldn't" as an answer.

Part 7: Open Methodology vs. Vendor-Captured Standards

There's a reason the PERA methodology is maintained as an open resource rather than a proprietary framework or a locked standard.

The Problem with Closed Standards

Standards bodies like ISA and IEC do important work. But participation requires membership fees, and published standards require purchase. The IEC 62443 series-essential for industrial cybersecurity-costs hundreds of dollars per document.

This creates a knowledge barrier. Small and medium facilities, the ones often most vulnerable, can't afford to participate in standards development or even to read the final documents. They're left with vendor interpretations and consultant summaries.

The Vendor Capture Problem

When standards require purchase to read, vendors become the interpreters. They attend the committee meetings, they understand the requirements, and they translate those requirements into product features.

This isn't malicious, but it does create systematic bias. Every vendor interprets standards in ways favorable to their products. Over time, the market's understanding of a standard becomes the vendor-filtered version, not the actual document.

The cartoon Purdue Model is a perfect example. The diagram vendors use to sell segmentation products bears only superficial resemblance to the actual PERA methodology.

Why Open Matters

PERA is freely available at pera.net. You can read the original documentation, examine the diagrams, understand the methodology, and apply it to your facility without paying licensing fees or depending on vendor interpretation.

This openness is philosophically important:

- **Practitioners can verify.** When someone claims their product "implements Purdue," you can check the methodology yourself.

- **Small operators can participate.** A rural water utility can apply the same framework as a multinational refinery.
- **Development continues in the open.** Updates, extensions, and applications are visible and reviewable.

The goal isn't to compete with IEC 62443 or other standards. The goal is to provide the conceptual foundation that helps practitioners understand and apply those standards effectively.

Conclusion: Stop Mapping Cables. Start Mapping Consequences.

The Purdue Model isn't dead. It was never fully alive in most practitioners' minds, because what they knew as "Purdue" was a simplified diagram stripped of its methodological context.

The actual PERA methodology - comprehensive, consequence-aware, and designed for exactly the kind of integration challenges we face today - remains available for anyone willing to engage with it.

The question isn't whether the hierarchy is outdated. The question is whether you're willing to do the consequence analysis that the hierarchy enables.

- When you connect a system to the network, what fails when that connection fails?
- When you grant remote access, what can be done and what are the physical consequences?
- When you integrate cloud services into operations, what Level 4 failures now cascade to Level 1?

These are PERA questions. They were always PERA questions.

The methodology that answers them is open, documented, and waiting.

pera.net

🌊

This article is part of ongoing work to restore understanding of the Purdue Enterprise Reference Architecture and its application to modern industrial cybersecurity challenges. The PERA methodology is maintained by a community of practitioners committed to open development and consequence-aware industrial design.